

20 LECCIONES DE BASIC
PARA APRENDER CON EL SPECTRUM

INGELEK

JACKSON

- Los periféricos del SPECTRUM
- Analizar el programa
- Los diagramas de bloques
- Escribir el programa
- INPUT, LIST, RUN
- Tu primer programa
- Los operadores aritméticos
- Videotest n. 2

2

Spectrum

16K/48K/PLUS

VIDEO BASIC
Una publicación de
INGELEK JACKSON
Director editor por INGELEK:
Antonio M. Ferrer
Director editor por JACKSON HISPANIA:
Lorenzo Bertagnolio
Director de producción:
Vicente Robles
Autor: Softidea
Redacción software italiano:
Francesço Franceschini,
Stefano Cremonesi
Redacción software castellano:
Fernando López, Antonio Carvajal,
Alberto Caffaratto
Diseño gráfico:
Studio Nuovaidea
Ilustraciones:
Cinzia Ferrari, Silvano Scolari,
Equipo Galata
Ediciones INGELEK, S. A.
Dirección, redacción y administración,
números atrasados y suscripciones:
Avda. Alfonso XIII, 141
28016 Madrid. Tel. 2505820
Fotocomposición: Espacio y Punto, S. A.
Imprime: Rotacolor
Reservados todos los derechos de reproducción y publicación de diseño, fotografía y textos.
©Grupo Editorial Jackson 1985.
©Ediciones Ingelek 1985.
ISBN del tomo 1: 84-85831-15-2
ISBN del fascículo: 84-85831-14-4
ISBN de la obra completa: 84-85831-13-6
Depósito Legal:
Plan general de la obra:
20 fascículos y 20 casetes, de aparición quincenal, coleccionables en 5 estuches.
Distribución en España:
COEDIS, S. A.
Valencia, 245. 08007 Barcelona.
INGELEK JACKSON garantiza la publicación de todos los fascículos y casetes que componen esta obra y el suministro de cualquier número atrasado o estuche mientras dure la publicación y hasta un año después de terminada.
El editor se reserva el derecho de modificar el precio de venta del fascículo, en el transcurso de la obra, si las circunstancias del mercado así lo exigen.
Mayo, 1985.
Impreso en España

INGELEK

JACKSON

SUMARIO

Introducción

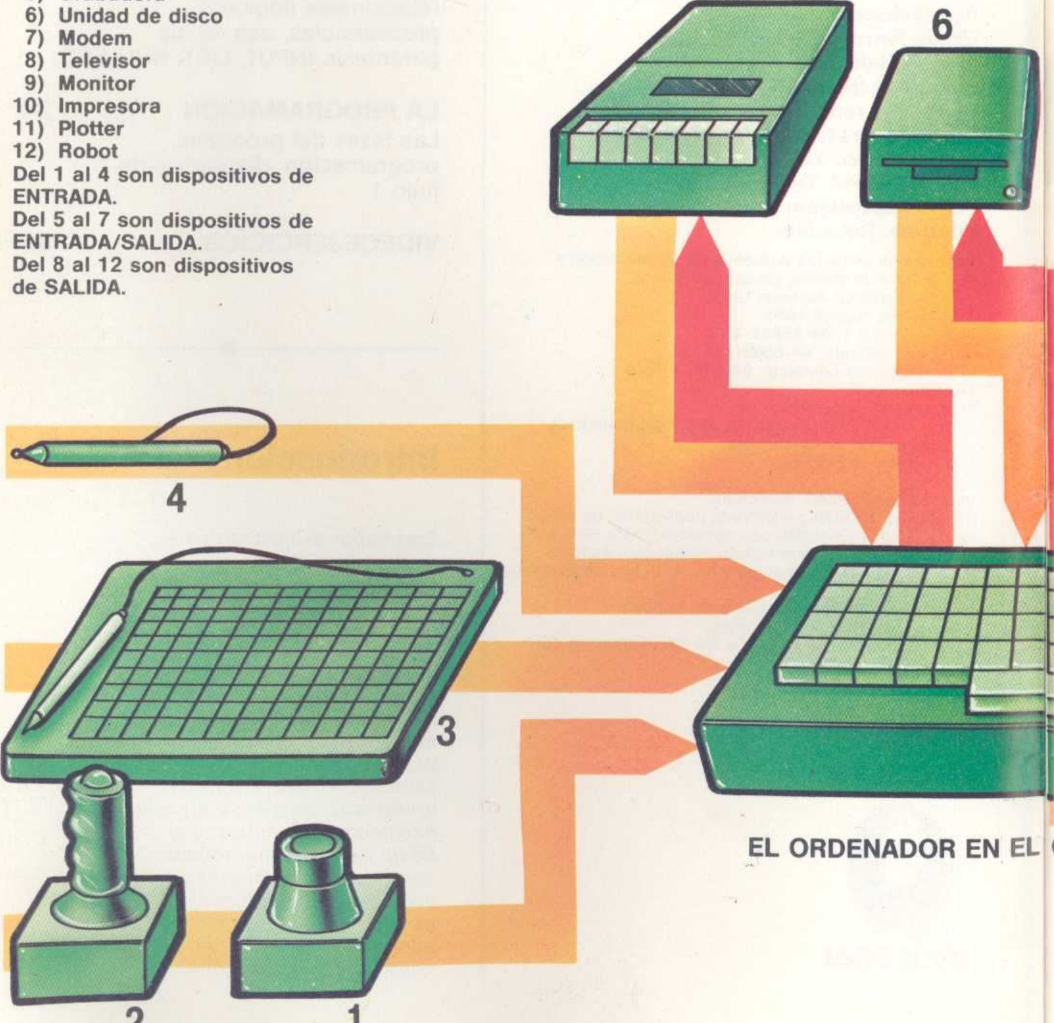
Tras haber examinado en la primera lección de VIDEOBASIC el esquema de un ordenador, en las próximas páginas trataremos de las unidades complementarias que pueden conectarse a tu Spectrum. En la parte dedicada a la teoría descubriremos cómo usa el BASIC algunos símbolos de operaciones aritméticas, distintos a los que habitualmente empleamos. Después, tras haber tomado contacto con las nuevas instrucciones y órdenes, estaremos por fin capacitados para proyectar y escribir nuestro primer programa.

Los periféricos del Spectrum

Como ya se ha dicho, la parte fundamental y más característica del Spectrum es la unidad central, cuyo cometido, similar al de la mente humana, es el de elaborar los datos. Pero son también necesarios dispositivos capaces de pasar esos datos a la «mente» del ordenador y otros más, que dejen disponibles y utilizables los resultados de la elaboración.

1) **Paddle**
2) **Joystick**
3) **Tableta gráfica**
4) **Lápiz óptico**
5) **Grabadora**
6) **Unidad de disco**
7) **Modem**
8) **Televisor**
9) **Monitor**
10) **Impresora**
11) **Plotter**
12) **Robot**

Del 1 al 4 son dispositivos de ENTRADA.
Del 5 al 7 son dispositivos de ENTRADA/SALIDA.
Del 8 al 12 son dispositivos de SALIDA.

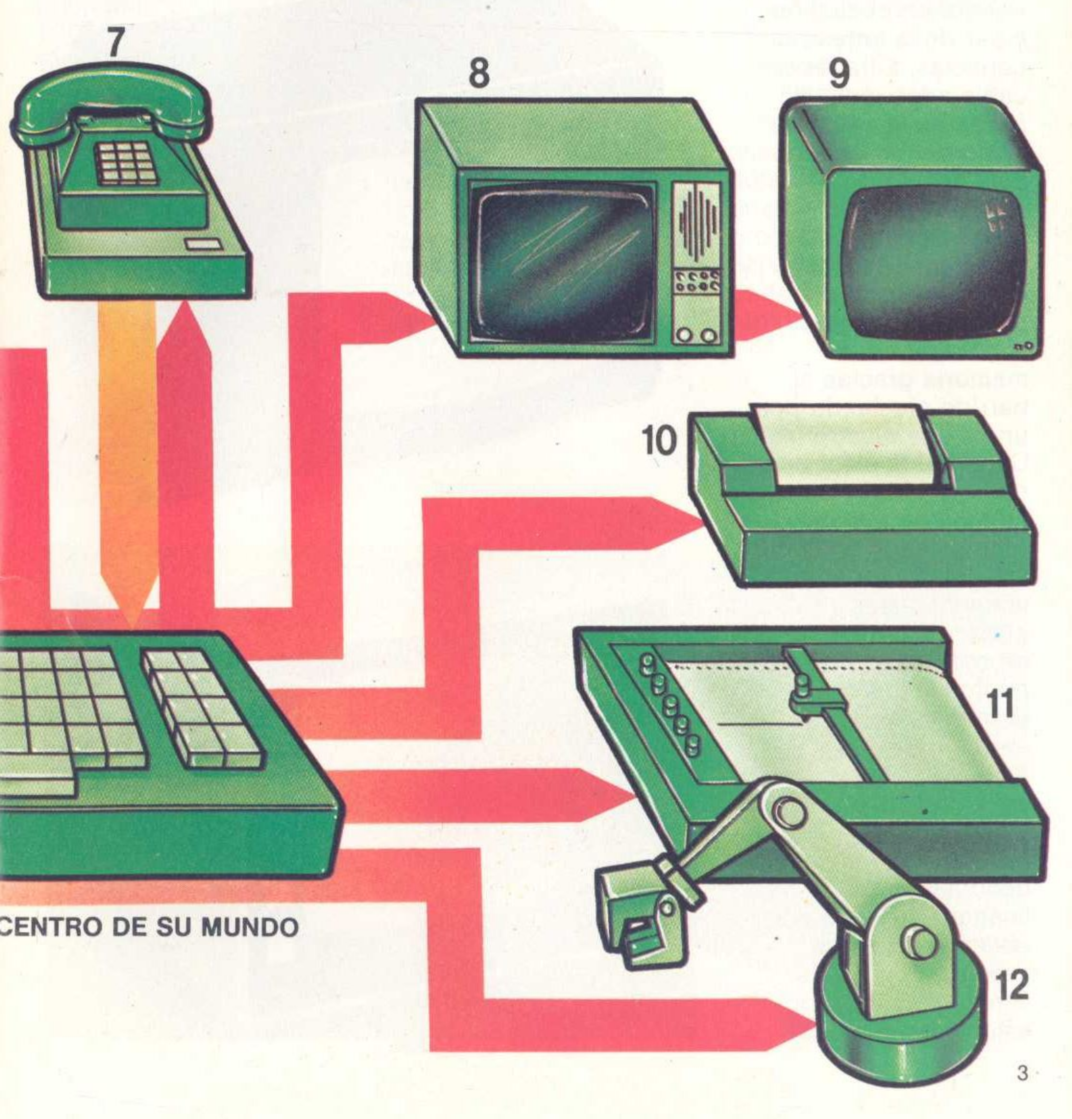


EL ORDENADOR EN EL

HARDWARE

Nuestro cerebro, aunque capaz de hacer miles de cosas, necesita del cuerpo para poder convertir en actos sus capacidades.
De la misma manera, para poder actuar la unidad central necesita «brazos» u «ojos»: las unidades periféricas; nombre debido al hecho de rodear, habitualmente a la unidad central, a la que casi siempre están físicamente unidas.

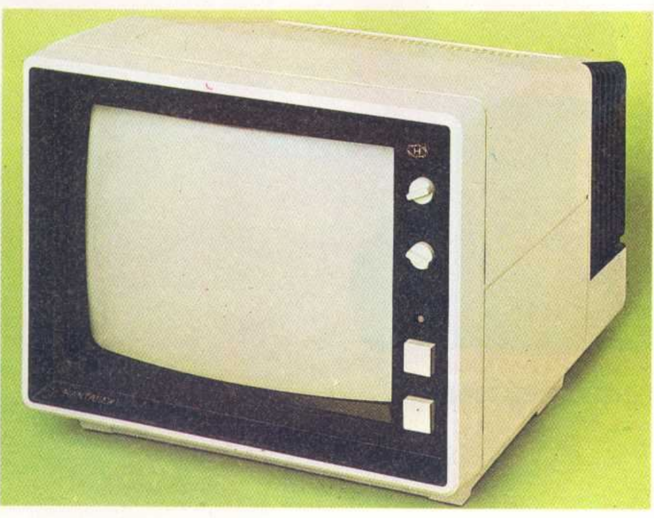
Pueden ser de ENTRADA, cuando transmiten datos a la CPU, de SALIDA cuando los reciben y de ENTRADA/SALIDA cuando el mismo dispositivo realiza ambas funciones.



TV y monitor

Tanto la TV como el monitor son dispositivos para visualizar la señal de vídeo que proviene del Spectrum.
La primera es un televisor doméstico normal en el cual en lugar de la antena conectas, a través del cable adecuado, la salida de la señal de vídeo del ordenador. Esta señal es adecuadamente decodificada (transformada) por la TV, que reproduce en pantalla la imagen contenida en la memoria gracias al barrido efectuado por un haz de electrones. Cuando los electrones colisionan con los átomos de fósforo (depositados en la cara interior del tubo de imagen), éstos emiten luz, reproduciendo mediante puntos la imagen.
El monitor es la evolución natural del televisor y, respecto a este último, se diferencia por el proceso de decodificación de las imágenes, con el que se obtiene una mejor calidad.
Gracias a su mayor estabilidad y nitidez, permite una lectura más descansada de las informaciones, y por esta razón es aconsejable para quien emplea el terminal de vídeo para usos profesionales. En cambio, para un uso lúdico del ordenador es más que suficiente una buena televisión.





Grabadora

Se emplea como memoria de masa, es decir como un contenedor donde se pueden almacenar, sin perderlas, una gran cantidad de informaciones.
Estas son convertidas en sonidos por el ordenador, según un patrón definido, y así son grabadas en cinta magnética, al igual que cualquier otra señal de audio.
La grabadora es uno de los elementos más delicados del sistema: si alguna de sus partes no funciona, no lograrás cargar las informaciones anteriormente memorizadas en cinta.
Por esta razón, es aconsejable poner el máximo cuidado en el uso de esta unidad.
Una norma particularmente importante es limpiar periódicamente las cabezas de grabación.
Evitarás así distorsiones que engañan al ordenador, enviándole informaciones equivocadas.
Es también aconsejable grabar a un volumen elevado, de modo que la señal sea fácilmente «audible» por el ordenador.
Por último es recomendable utilizar el cuentavueltas y tomar nota de los nombres y posiciones de los programas existentes en el casete, para así poderlos localizar rápidamente.



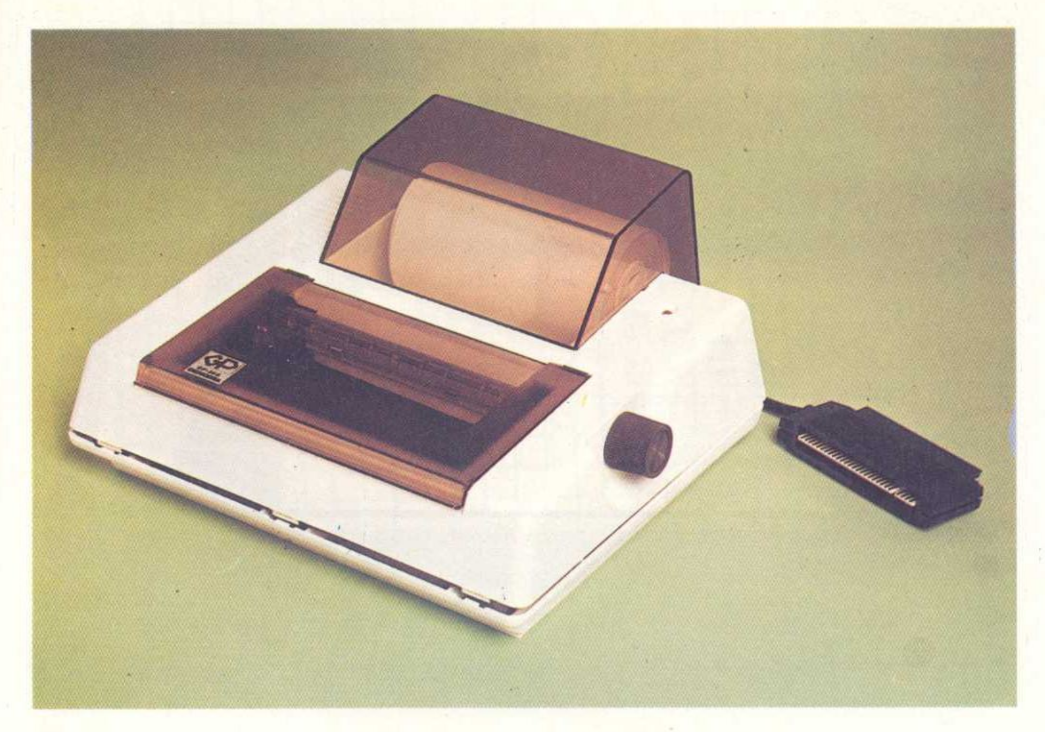
Unidad de disco

Esta unidad desempeña la misma tarea que la grabadora y por lo tanto los discos flexibles son también memorias de masa.
La principal característica que diferencia a las unidades de disco de las grabadoras es la velocidad con la que se cargan y memorizan las informaciones.
En efecto, la velocidad de carga varía de un mínimo de 2 a un máximo de 8/10 veces la de la grabadora.
Las dimensiones de los discos, o disquetes como también se les llama, más usuales es de 5 1/4 pulgadas; sin embargo existen otros formatos (3", 3 1/2", etc.), dependiendo de la unidad de disco empleada.
Otra importante diferencia respecto a la grabadora reside en el método de búsqueda de una información.
En los disquetes, es posible la búsqueda tanto hacia atrás como hacia delante, puesto que el cabezal puede desplazarse radialmente, al contrario de lo que ocurre en la grabadora, donde es la cinta quien pasa delante de la cabeza, que está fija.
Se dice que el disco es una memoria de masa de acceso aleatorio, porque permite la búsqueda del dato directamente en la posición en la que ha sido memorizado.
La cinta, en cambio, por el hecho de desplazarse únicamente hacia delante mientras carga los datos, es llamada memoria de masa secuencial.
En efecto, sobre la cinta las informaciones se memorizan la una a continuación de la otra, es decir, secuencialmente.
Otro tipo veloz de memoria de masa, pero típico y exclusivo del fabricante del Spectrum, es el microdrive.
Es fundamentalmente una grabadora muy rápida que utiliza unos pequeños cartuchos que contienen la cinta, la cual tiene el final unido directamente con el principio, por lo que ya no son necesarias las operaciones de parada del motor y de rebobinado.
De este modo, la lectura o escritura de los datos se realiza a una velocidad mucho más alta que en una grabadora e inferior únicamente a la de una unidad de disco. Tanto la unidad de disco como el microdrive se conectan al Spectrum a través del un específico circuito electrónico externo llamado interface.



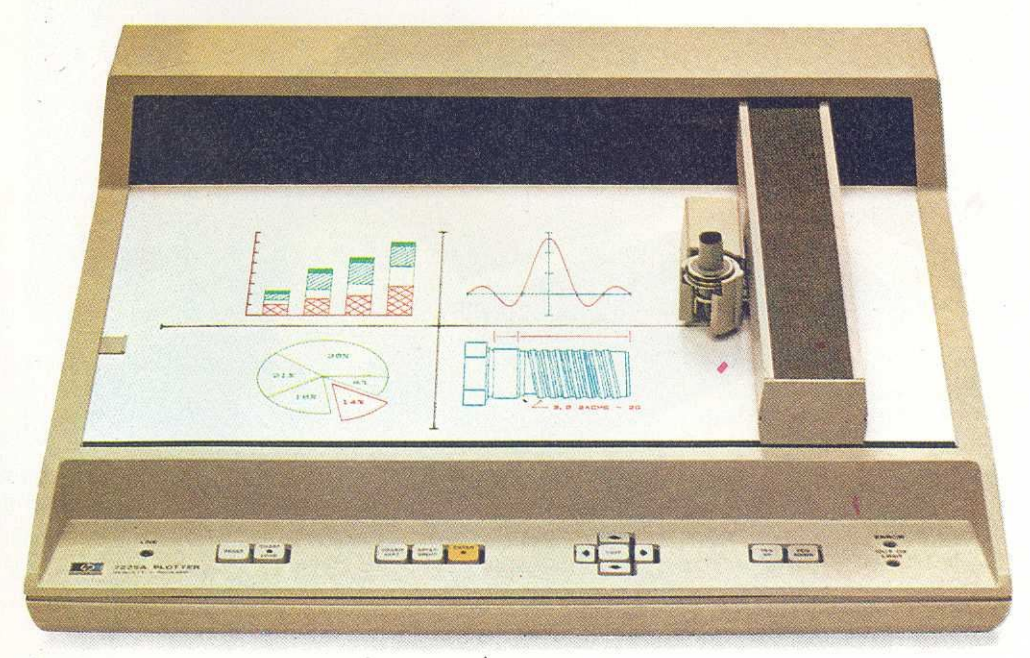
Impresora

La impresora es un dispositivo capaz de imprimir sobre papel las informaciones elaboradas (números y textos).
En la inmensa mayoría de los casos, su funcionamiento es básicamente similar al de una máquina de escribir: un cabezal imprime el carácter deseado sobre una cinta entintada en contacto con la hoja.
Las impresoras más difundidas adoptan cabezales con agujas, las cuales, empujadas por martilletes, forman el carácter a imprimir mediante puntos.
En las impresoras de margarita, en cambio, los caracteres están ya físicamente presentes, exactamente igual que en una máquina de escribir.
Las impresoras de agujas en general son más rápidas (pueden imprimir desde 30 hasta más de 400 caracteres por segundo) y relativamente económicas.
Las de margarita son más lentas (de 10 a 50 caracteres por segundo) y más caras, pero en cambio consiguen una mejor calidad de impresión.



Plotter

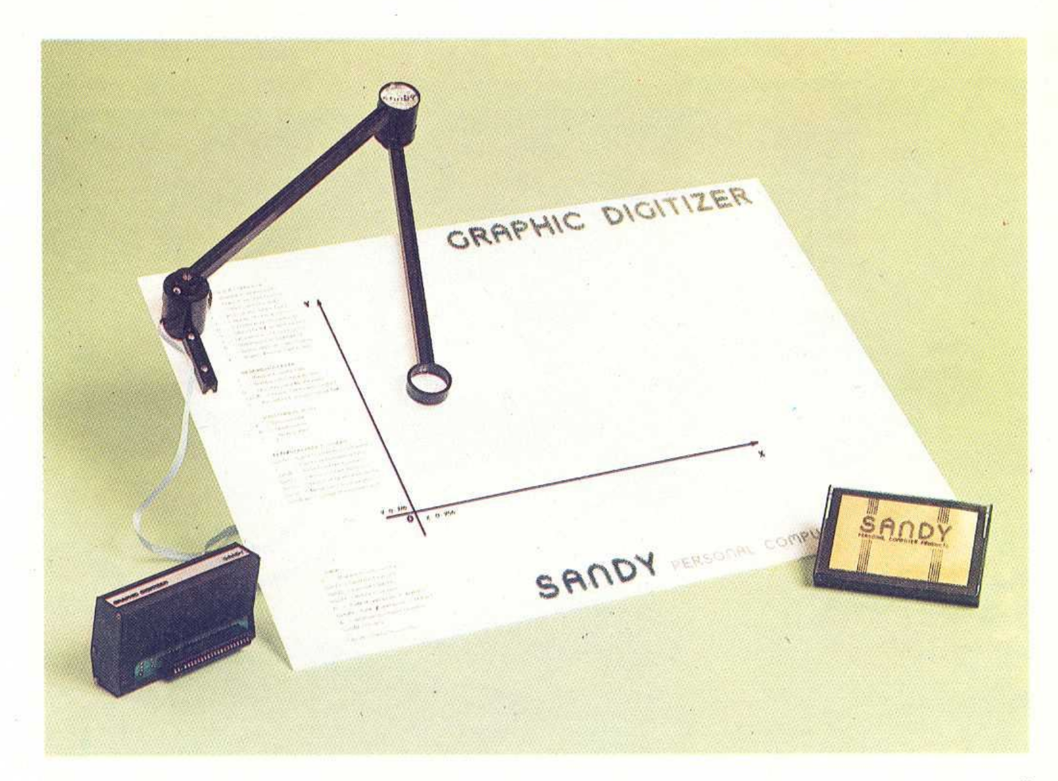
Las impresoras tienen algunas limitaciones en el uso gráfico: precisión no demasiado elevada (la imagen está constituida por puntos aislados) y dimensiones limitadas de la hoja Cuando sea en cambio necesario imprimir sobre papel o película de acetato, se recurre al plotter (o trazadora de gráficos). Se trata en esencia de un tecnígrafo automático, consituido por un plumín que se desliza a lo largo de dos guías cuyo desplazamiento es, a su vez, libre.
Según el plumín esté subido o bajado trazará un punto si las guías están paradas y una línea si están en movimiento. De esta sencilla manera el plotter puede realizar dibujos muy complejos. Y no solamente esto. El plotter puede escribir textos como cualquier impresora, pero sin el límite que constituye el tipo de letra del cabezal; es capaz de imprimir cualquier carácter en cualquier estilo.



Tableta gráfica

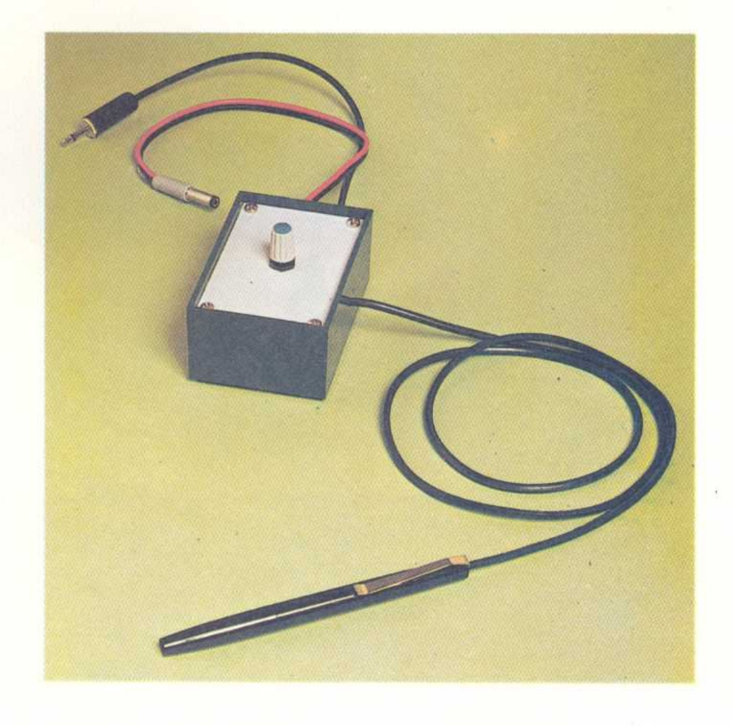
Es una especie de pizarra electrónica donde la tiza la constituye una pluma especial y la pizarra un plano sensible a la acción de la pluma. Es, en otras palabras, el recíproco del plotter: se trata de otro tecnígrafo, pero destinado a comunicar informaciones al ordenador, y no a visualizar aquéllas recibidas desde la máquina. Su uso es el mismo de una pizarra tradicional, pero con numerosas ventajas respecto a ella: posibilidad de borrar con facilidad, todo o en parte, lo que se haya hecho; posibilidad de poder construir figuras geométricas con la ayuda del ordenador, simplemente posicionando la pluma de un manera particular; posibilidad de corregir muy sencillamente cada dibujo; posibilidad de colorear un área determinada.

La tableta gráfica (o digitalizador) es un instrumento preciso para cualquiera que se interese profesionalmente por cuestiones gráficas (ingenieros, arquitectos, gráficos, estilistas...); por su sencillez y potencia deja atrasadas a las anteriores mesas de dibujo.



Lápiz óptico

El lápiz óptico es el análogo para la pantalla de la pluma de la tableta gráfica: junto con ella tiene, más o menos, las mismas funciones del digitalizador con (en referencia a éste) una menor precisión, pero una mayor economía.
El principio de funcionamiento del lápiz óptico, o light pen, es bastante sencillo. Recordarás que la imagen de un televisor o monitor está constituida por puntos luminosos debidos a la excitación del fósforo. Cuando la pluma, en cuya extremidad está montado un sensor óptico, capta luminosidad, quiere decir que el haz de electrones está incidiendo directamente sobre los átomos de fósforo que están inmediatamente debajo. En este momento, circuitos especiales determinan las coordenadas exactas del punto sobre el que está apoyada la pluma.



Joystick

El joystick (una especie de elemental teclado externo) es un dispositivo de entrada constituido esencialmente por una palanca orientable al menos en cuatro direcciones (arriba, abajo, derecha, e izquierda), con la cual puedes suministrar sencillas informaciones de posición al ordenador.
Conocido y usado por todos los apasionados de los telejuegos, permite desplazar de forma inmediata un elemento determinado a través de la pantalla.
Otra ventaja respecto al teclado tradicional es la posibilidad de unir a una orden, como por ejemplo subir el cursor, la acción física de desplazar con la mano la palanca hacia arriba, haciendo fácil y natural, además de cómodo, el uso de esta unidad. La única advertencia es que no todos los joysticks son conectables a los ordenadores domésticos, por lo tanto es aconsejable informarse sobre cuales pueden emplearse en tu Spectrum. Por su comodidad es un accesorio que aumenta notablemente la economía del sistema (es decir, la facilidad de manejo por parte del usuario).



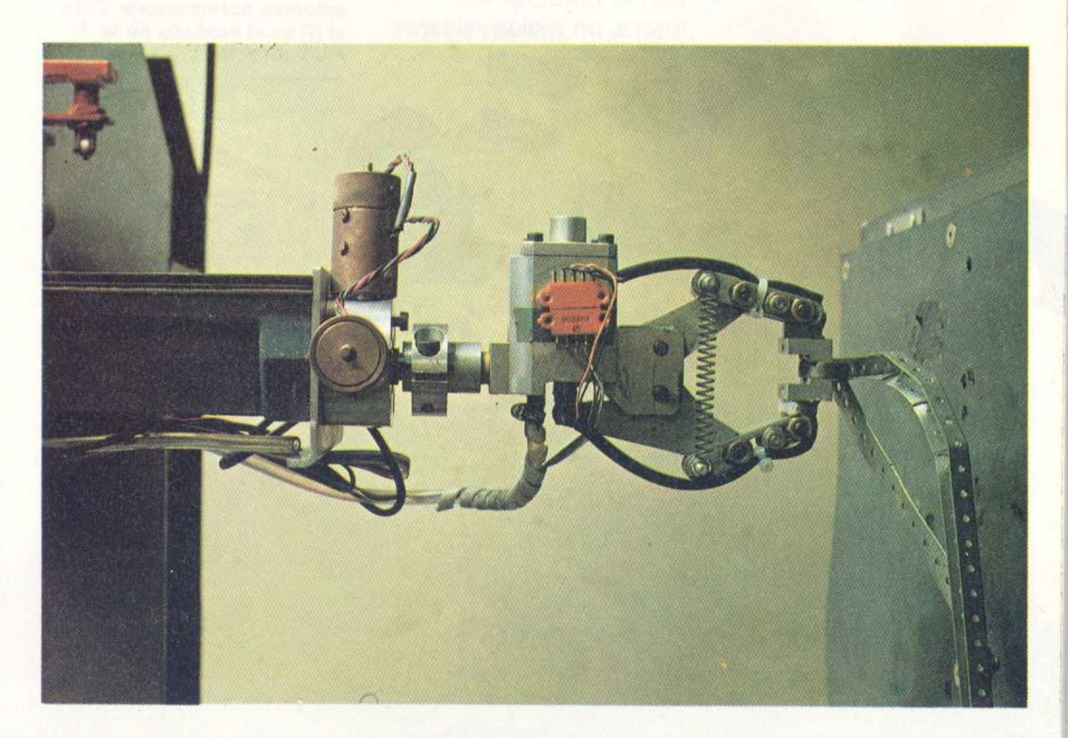
Modem

Es la abreviatura de modulador/ demodulador, es decir, una unidad capaz de convertir datos y programas a la forma necesaria para poder ser transmitidos a través de cable telefónico, y, naturalmente, capaz de realizar la operación opuesta: «traducir» las señales provenientes de la línea telefónica en datos utilizables.
Por lo tanto, permite conectar un ordenador a otro, poniendo en común datos y programas también a través de distancias considerables en brevísimo tiempo.
Sobre este principio se han fundamentado los bancos de datos, enormes archivos electrónicos que ponen informaciones de todo tipo a disposición del usuario.
Por ejemplo, puedes conocer todos los libros sobre un tema determinado publicados en España, o alquilar una habitación quedándote tranquilamente en casa.
Un tipo de modem de bajo coste es el llamado acoplador acústico, utilizable en todos los ordenadores domésticos y personales.
Una vez conectado a tu Spectrum, su uso es muy sencillo: basta con apoyar el auricular telefónico en el alojamiento de la unidad, disponer el sistema para transmitir o recibir datos, marcar el número telefónico que corresponda al ordenador a llamar, y finalmente, transmitir (o recibir) datos.



Robot

El nombre Robot (derivado del checoslovaco Robota) fue acuñado por el comediógrafo A. Capevi para definir, en el ámbito de una de sus comedias, a unos servidores mecánicos. La ciencia-ficción, posteriormente, se ha adueñado del término para indicar con él máquinas de aspecto y comportamiento humanos.
En realidad, un robot es una unidad electrónico-mecánica (bien distinta del nombre), gobernada por un ordenador, y capaz de cumplir determinadas acciones (trasportar materiales, montar piezas mecánicas, aplicar pinturas...) en sustitución del hombre en actividades peligrosas, repetitivas, o de extremada precisión.
En el caso de un robot para ordenadores domésticos, como tu Spectrum, la mayoría de las veces su aspecto es el de un brazo mecánico capaz de realizar (por ahora) sencillas operaciones, gracias a los programas adecuados implementados en el ordenador.
Para concluir, el robot es la prolongación «física» del ordenador hacia el mundo exterior, lo que le permite convertir en acciones los datos elaborados.

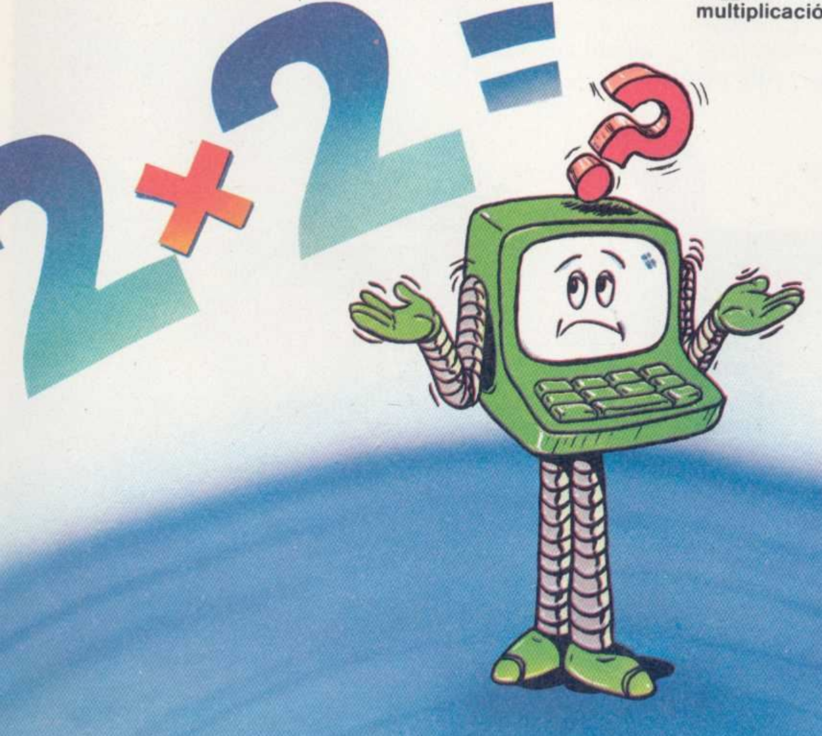


Operadores aritméticos

Las expresiones matemáticas, en BASIC, son prácticamente idénticas a las que has estudiado en el colegio, con algunas pequeñas diferencias en la escritura de los símbolos de las operaciones.
En BASIC las operaciones de suma y resta se indican como de costumbre con «+» y «—».

El producto en cambio se indica con el «*» y no con «x» o el punto.
En matemáticas, con frecuencia, el producto de dos valores se indica escribiéndoles sencillamente uno detrás de otro; así «a b» significa «a por b». En cambio, para tu Spectrum nunca queda nada sobreentendido, por lo que el producto ha de ser expresamente indicado con el asterisco: A * B (nota que en BASIC todo va en letras mayúsculas).
La división se indica con el carácter «/» (barra, en inglés «slash»); los dos puntos (:) serían interpretados por tu Spectrum como signo de puntuación.
La elevación a potencia es otra operación con un símbolo particular. No se escribe el exponente como un número más pequeño situado arriba, a la derecha de la base como «4^3». Se escriben en cambio base y exponente en la misma línea, separados por «↑», para nuestro ejemplo 4 ↑ 3.

Tu Spectrum no reconoce algunos de los símbolos que empleas normalmente. Para el [*] es el símbolo de la multiplicación.



Los operadores relacionales

Los operadores de relación son símbolos que sirven para establecer órdenes de magnitud entre dos elementos. Los más comunes son =; ≠; >; <.

a) El símbolo = representa que dos o más magnitudes están ligadas por una relación de identidad. Por ejemplo:

□ = □;
4 = 4;
4 = 4
4 igual a 4

b) Su contrario es la relación de desigualdad, representada por el símbolo matemático ≠, relación que es mucho más fácil de identificar, en un plano estrictamente numérico, que la anterior. En efecto, son infinitamente más numerosas las cantidades desiguales entre ellas que las iguales.

□ ≠ △

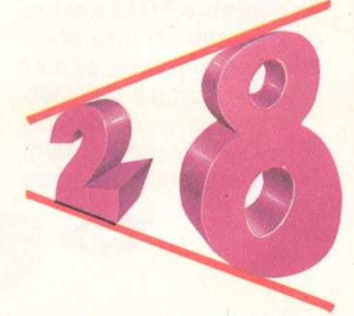
2 ≠ 3

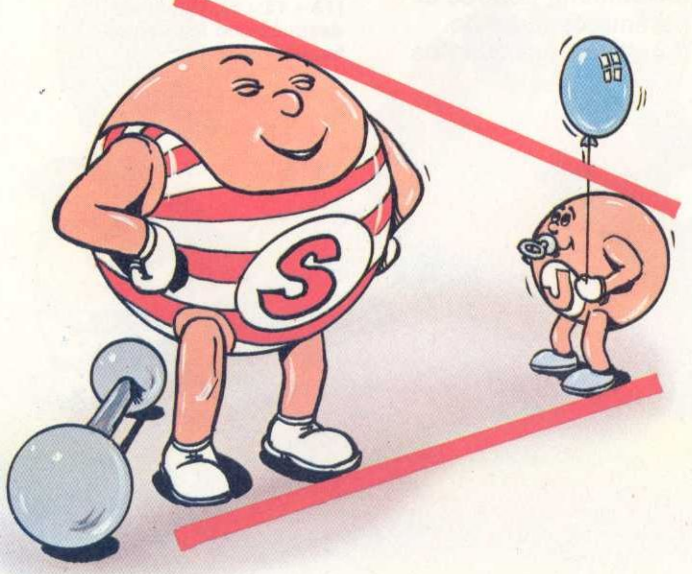
2 es distinto que 3

c) Dos magnitudes distintas pueden a su vez ser mayores o menores la una respecto de la otra. Estos otros dos tipos de relación se expresan con los símbolos mayor (>) y menor (<).

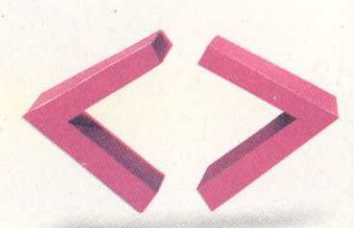
6 > 2
6 es mayor que 2

2 < 8
2 es menor que 8





Nota. Para el ordenador el concepto de desigualdad se representa gráficamente con el símbolo < >.



LENGUAJE

Los paréntesis y su prioridad

En matemáticas existen símbolos, llamados paréntesis, empleados en el cálculo de expresiones, aritméticas o algebraicas, que pueden encerrar algunos elementos de dichas expresiones.
Las operaciones entre paréntesis se realizan antes que las demás.
Existen tres tipos de paréntesis que determinan el orden de ejecución de las operaciones y son: el paréntesis redondo (), el paréntesis cuadrado [], y las { } .
Estos indican en su orden de sucesión la prioridad de una operación simple.
El BASIC, en cambio, únicamente conoce el paréntesis redondo.
Puedes poner cuantos quieras, ¡lo importante es cerrarlos!
Tu Spectrum empezará a calcular a partir del paréntesis redondo situado más al interior de la expresión, para continuar sucesivamente hacia el exterior.
Resumimos ahora en orden de prioridad los operadores algebraicos y de relación:

1) elevación a potencia;
2) multiplicación - división;
3) suma - resta;
4) igual - mayor - menor.

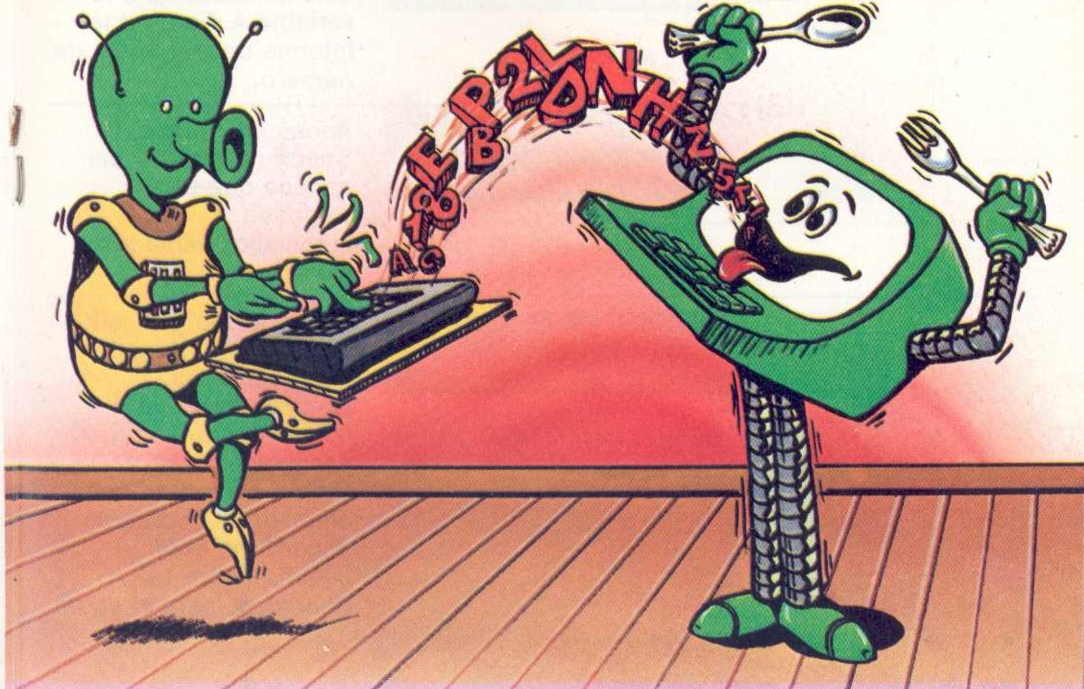
Tu Spectrum empieza a calcular desde el paréntesis más interior, resolviendo primero la expresión (18 - 12 - 4), siguiendo después con los demás hacia el exterior.

$$12 : (3 - 2 \times (6 : 2 + 4 \times (18 - 12 - 4) - 2) : 2) + 14$$

INPUT

Muy frecuentemente existen casos en los que al escribir el programa no se conocen los datos que habrán de ser elaborados.
Supongamos, por ejemplo, que tu próximo programa tenga por objetivo calcular la edad media de las primeras 5 personas que lo empleen.
Deberá, por lo tanto, operar sobre datos (la edad de esas personas) que te son desconocidos.

La única forma razonable de entrar en posesión de los datos necesarios es la de pedirlos directamente a quien pueda comunicarlos, es decir, a esas personas.
Esta es la tarea de la instrucción INPUT.
A través de ella puedes pedir al usuario del programa la introducción por el teclado de uno o más datos; y estos serán memorizados en la o las variables que aparezcan en la instrucción.
Cuando el programa encuentra la instrucción INPUT, tu Spectrum imprime un cursor parpadeante en la parte inferior de la pantalla.
Desde este momento cualquier carácter tecleado será impreso hasta que se pulse la la tecla ENTER, que pone fin a la fase de introducción de los datos.
Si los datos a comunicar son más de uno, es decir, si hay más variables en la instrucción INPUT, estos pueden introducirse seguidos, siempre que pulses ENTER detrás de cada uno de ellos.
Debes comunicar un dato por cada variable contenida en la instrucción, y el dato, cosa muy importante,



debe de ser del mismo tipo de la variable: numérica o de cadena. Si el dato es de tipo texto y la variable es numérica, aparecerá el mensaje:

```
variable not found
```

El Spectrum te informa de que está en espera de una cadena y no de un número cuando el cursor parpadeante aparece entre dos comillas.
Puesto que el programa puede requerir muchos datos de entrada, es indispensable que tu Spectrum te indique a ti (o a quien esté usando el programa) qué datos se deben insertar.
De ahí la necesidad de anteponer un mensaje entre comillas, a la o a las variables –la primera de las cuales irá precedida por ; (punto y coma)–, contenidas en la instrucción INPUT.

```
INPUT A
```

"¿Qué significa la variable A?: ¿la fecha de nacimiento, la edad, el código fiscal?..."
Para evitar estos dilemas es necesario escribir:

```
INPUT "EDAD"; A
```

Así sabes y sabrás siempre, que el dato a introducir en la variable A es la edad.
Veamos algunos ejemplos del empleo de la instrucción INPUT:

Ejemplos:

```
INPUT A
```

Tu Spectrum espera de ti un número que será asignado automáticamente a la variable A. Pero no te informa de qué clase de número.

```
INPUT A$
```

Ahora, es cambio, tu Spectrum espera una cadena que será asignada automáticamente a la variable A$.

```
INPUT "NOMBRE DE CIUDAD"; V$
```

La respuesta que has de darle al ordenador es una cadena constituida por el nombre de una ciudad, por ejemplo, MADRID. Queda entendido que un error de escritura (Madrd) no será detectado por tu Spectrum: para él es un

texto más que aceptable.
Lo mismo sucede con los números. Si le comunicas un valor equivocado, o mejor dicho, distinto del que hubieras deseado introducir, tu Spectrum hará sus elaboraciones sobre éste, entregándote al final un resultado equivocado.
Presta atención por lo tanto a la introducción correcta de los datos.

```
INPUT "DIME 3 NUMEROS"; A, B, C
```

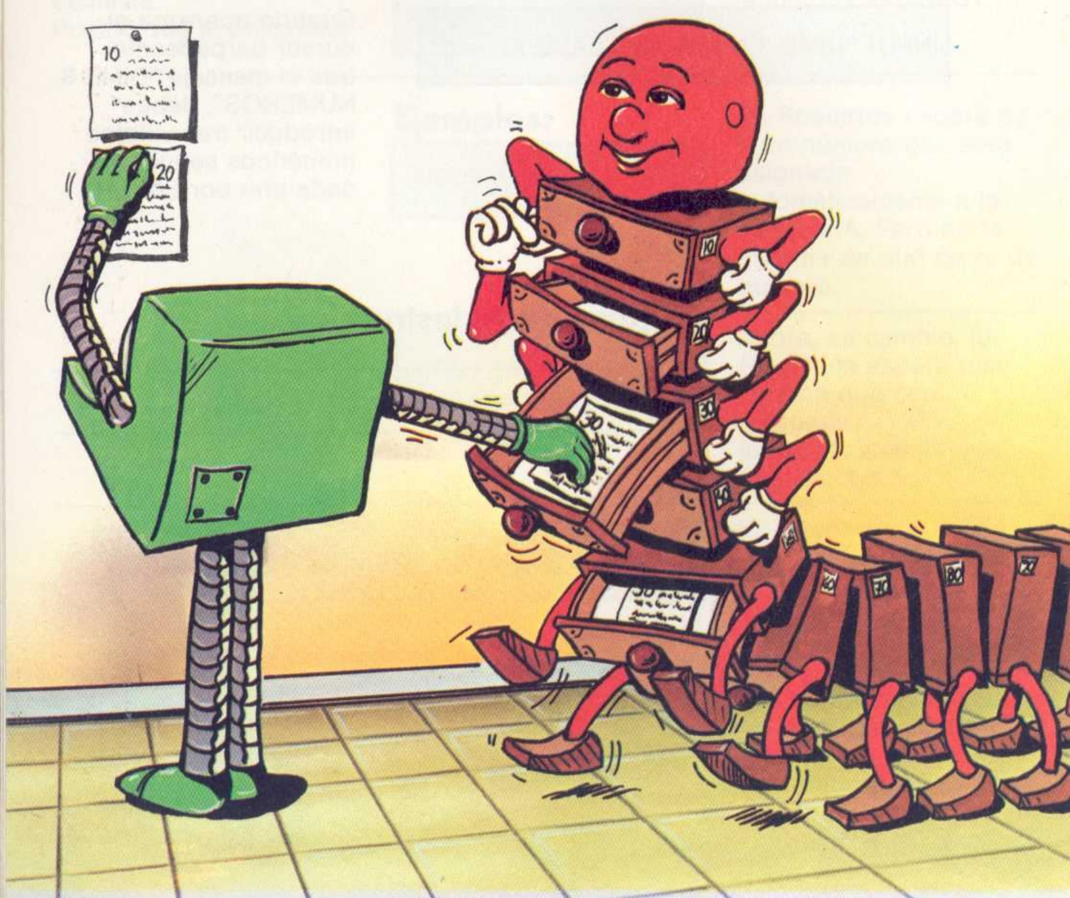
Cuando aparezca el cursor parpadeante, tras el mensaje "DIME 3 NUMEROS", debes introducir tres valores numéricos seguidos cada uno por ENTER.

Sintaxis de la instrucción

INPUT ["Mensaje";] variable [{' ,} variable ...]

LIST

Visualiza en pantalla el programa presente en ese momento en la memoria central del Spectrum.
El texto del programa, es decir, el conjunto de todas sus líneas escritas en lenguaje BASIC, toma el nombre de listado.
La instrucción LIST permite, en efecto, listar un programa.
Los programas BASIC siempre tienen sus líneas numeradas, por lo que es posible visualizar todo el programa o una parte de él, indicando la primera línea desde la que parte el listado.
LIST sin ninguna indicación de línea, lista todo el programa.
Indicando un solo número de línea al Spectrum, este lista a partir de esa línea.
LIST se usa normalmente en modo inmediato cuando se quiere controlar qué programa está presente en la memoria central, o bien, cuando se está escribiendo y modificando un nuevo programa. Por esta razón a LIST sería más correcto llamarlo "comando".



LIST podría emplearse también en modo diferido, en el interior de un programa, haciendo listar automáticamente el propio programa en el momento de su ejecución.
Si el listado que deseas visualizar es más largo de lo que cabe en pantalla, tu Spectrum te presenta la primera pantalla, e imprime abajo el mensaje «scroll?».
Si quieres bloquear el listado, pulsa la tecla N, o bien, STOP (pulsación simultánea de SYMBOL SHIFT y A) o bien BREAK; si quieres, por el contrario, pasar a la pantalla siguiente, basta con pulsar cualquier otra tecla.

Ejemplos

```
LIST 10
```

El Spectrum lista el programa de la memoria central, desde la línea 10 en adelante; o en el caso de que el programa no contenga la línea número 10, desde la primera línea siguiente, parándose en las primeras líneas visualizables en pantalla. Si quieres continuar con la lectura del teclado debes pulsar una tecla cualquiera, siempre que no sea N, STOP, o BREAK.

```
LIST
```

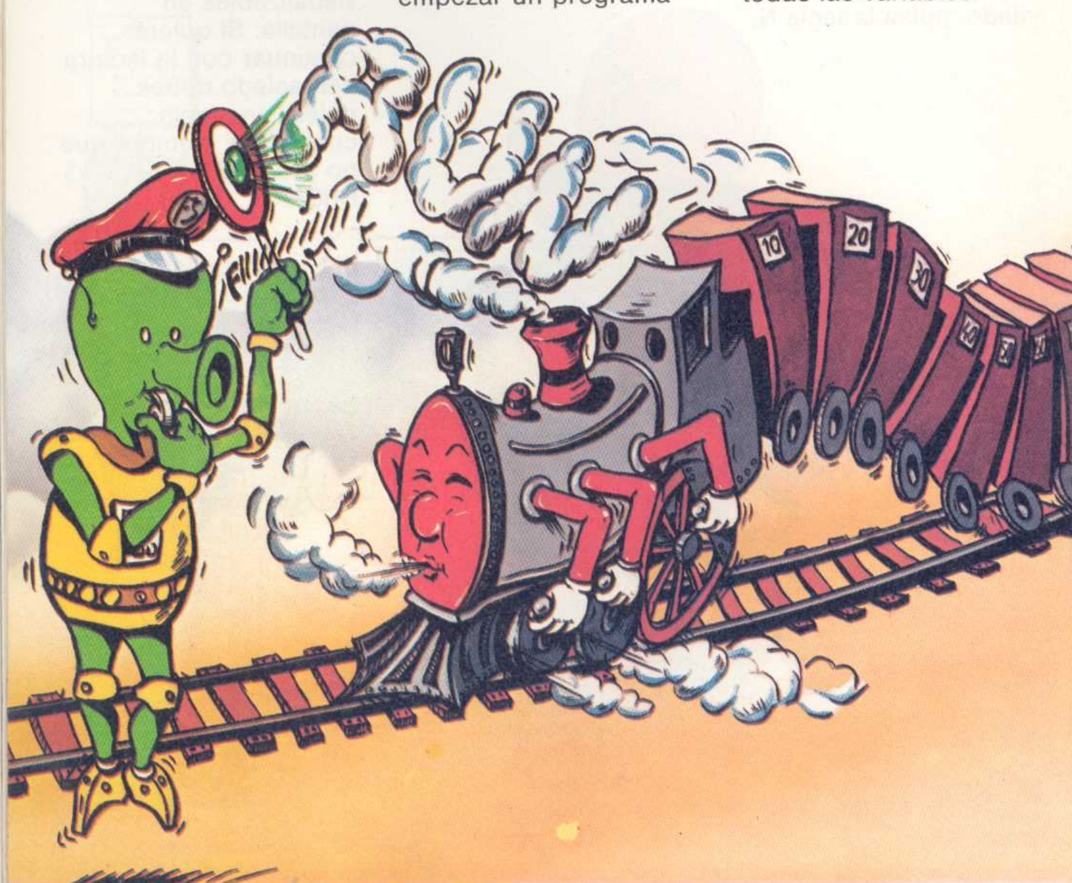
El Spectrum lista todo el programa a partir de la primera línea.

Sintaxis de la instrucción/comando

LIST [línea]

RUN

Instrucción del BASIC que pone en marcha el programa presente en la memoria central. Sin embargo, es más oportuno llamarla comando, dado que se da normalmente en modo directo sin estar precedida de número de línea.
La indicación de número de línea es opcional. Si falta, como es el caso más frecuente, el programa se ejecuta desde la instrucción con el número más bajo, cualquiera que sea este. Si se indica en cambio un número de línea, el programa empieza desde ésta. Sin embargo, en este caso, tiene que haber una razón lógica para hacer empezar un programa desde un punto distinto al del principio.
La instrucción RUN borra (hace RESET) cualquier posible valor de las variables previamente asignado.
En otras palabras, si se intenta reemprender con RUN línea (línea es el número de línea desde el que seguir), la ejecución de un programa parado por cualquier razón, se pierde el contenido de todas las variables.



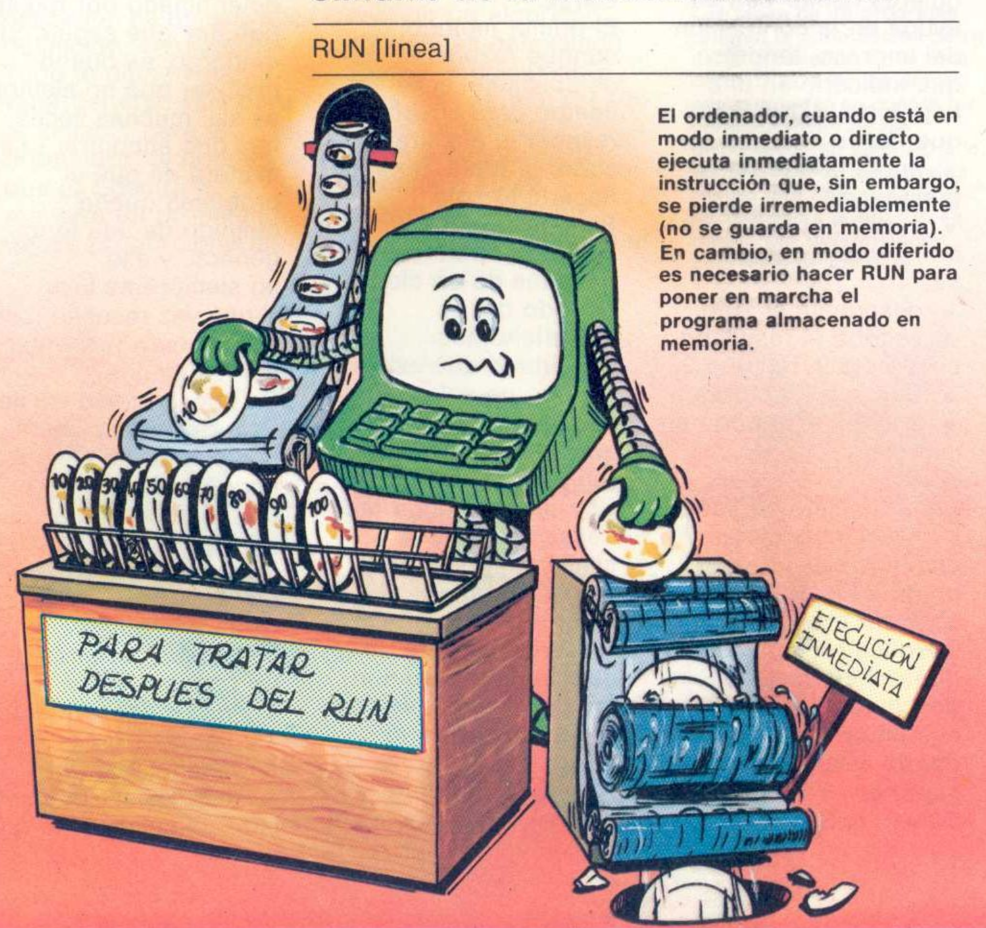
Ejemplos:

RUN	La ejecución del programa que está en la memoria central arranca desde la primera línea.
RUN 140	La ejecución del programa arranca desde la línea 140, cualquiera que sea su posición dentro del listado.

Sintaxis de la instrucción/comando

RUN [línea]

El ordenador, cuando está en modo inmediato o directo ejecuta inmediatamente la instrucción que, sin embargo, se pierde irremediablemente (no se guarda en memoria). En cambio, en modo diferido es necesario hacer RUN para poner en marcha el programa almacenado en memoria.



Algunos ejemplos elementales

Supón que sepas hacer tu solo la declaración de la renta, pero que no tengas tiempo. Supón también que tengas un amigo que no tiene la más mínima idea del I.R.P.F, bienes gananciales y retenciones..., pero que disponga de todo el tiempo necesario. Si quieres contar con su ayuda en la confección del impreso, tendrás que indicarle en una hoja todo lo que tiene que hacer:

- anota lo que he ganado;
- escribe el resultado en la hoja 5, renglón 12;
- descuenta el 10%
- escribe el resultado en la hoja 2, renglón 9;
- Busca las facturas;
- Escribe el importe en la hoja 3;
- etc.

Pero tu amigo no tiene las ideas muy claras y empieza por preguntarse adonde ir a comprar el impreso. Después no entiende que quiere decir descontar, tiene que encontrar las facturas y donde está marcado el importe.
Tú y tu amigo habláis dos lenguas distintas, y lo que a uno le parece fácil (porque tiene los conocimientos necesarios) para el otro no lo es en absoluto.
Sin embargo, a pesar del fracaso inicial, buscáis una solución.
Sólo hay dos maneras de resolver el problema:
1.° enseñar al amigo todo lo que no sabe;
2.° intentar guiarle de forma que las instrucciones le sean claras.
El primer método, aunque razonable, no es aplicable: si tienes tiempo para enseñarle a rellenar la declaración sobre la renta, podrías hacerlo tú solo.
Escoges pues la segunda opción, y después de un cierto periodo de malentendidos, descubres que este método no solamente es más práctico, sino también mucho más rápido.
Con un ordenador las cosas son muy similares: basta con cambiar «amigo ignorante» por «ordenador» para darte cuenta de cuáles son los problemas que cada programador debe saber afrontar.
Hemos visto que el problema de base es el de hacerse entender: ¿cuáles son entonces, en general, las operaciones que deberemos hacer para «instruir» al ignorante ejecutor?
Lo primero que deberás hacer es asegurarte de cuál es el problema a resolver.
En nuestro ejemplo, la declaración sobre la renta, tenemos un problema bien definido y rígidamente determinado por reglas que hay que seguir. Sin embargo, es bueno precisar que no siempre es así: muchas veces hay que encontrar una manera de que el problema quede definido de una forma general, y eso no siempre es fácil.
Y una vez resuelto, ¿de qué manera presentar los resultados?
En nuestro caso existe una sola: cumplimentar correctamente el impreso.
Pero existen otros problemas que tienen más de una solución.
La respuesta a todas estas preguntas se llama algoritmo: un conjunto de instrucciones con las siguientes características:

- debe existir una sola interpretación de lo que

ordenan.

- han de llevar a un resultado definido: el producto del algoritmo;
- han de ser autosuficientes: han de bastar por sí solas para llevar el algoritmo a término;
- deben de estar implícitamente definidas: cada orden debe ser elemental (no divisible en partes ulteriores) y el significado de cada operación ha de ser, por definición, verdadero.

Todo lo dicho tiene valor general: vale para tu amigo y para el ordenador. Si quieres que tu Spectrum resuelva un problema, debes proporcionarle instrucciones con las características enunciadas.
Gracias a ésta, tu ordenador será capaz de resolver una tarea cualquiera, bajo la forma de instrucciones elementales,
bien definidas y fácilmente ejecutables.
Pero si tuvieras que hacerlo siempre así (encontrar todas las instrucciones elementales) para poder escribir un algoritmo emplearías demasiado tiempo. Precisamente para esto se han creado los lenguajes de programación: para permitir a quien programa usar una «lengua» sintética que el ordenador traduce automáticamente a instrucciones elementales (lenguaje máquina), que él es capaz de realizar.
Como cada lengua, cada lenguaje de programación (incluido el BASIC, del que nos ocupamos en este curso) tiene su gramática y sus sintaxis.
En primer lugar

Las fases de un programa reflejan la constitución física de tu ordenador. En la fase de ENTRADA de los datos hay que tener en cuenta las características de los dispositivos de ENTRADA, la elaboración de los datos la realiza una Unidad Central de Proceso y la SALIDA se realiza por determinado periférico de SALIDA.



contiene afirmaciones imperativas de forma que no resulten ambiguas y queden asociadas a una acción bien definida.
Contiene después reglas que prescriben de qué forma y en qué orden se deben escribir.

Los diagramas de bloques

Lo más difícil, una vez encontrado el algoritmo (la descripción completa y cuidadosa de los pasos necesarios para la resolución de un problema), es conseguir verificar rápidamente su validez.
Por esto nos valemos de una representación gráfica del algoritmo llamada diagrama de flujo o de bloques, en inglés flow chart. La simbología empleada es la siguiente: principio/final del algoritmo; operación de elaboración de datos; operación de elección en base a una situación; operación de entrada/salida.
Hasta con un lenguaje que nos releve del trabajo de traducción al lenguaje de la máquina (llamado precisamente

PRINCIPIO FINAL

ELABORACION

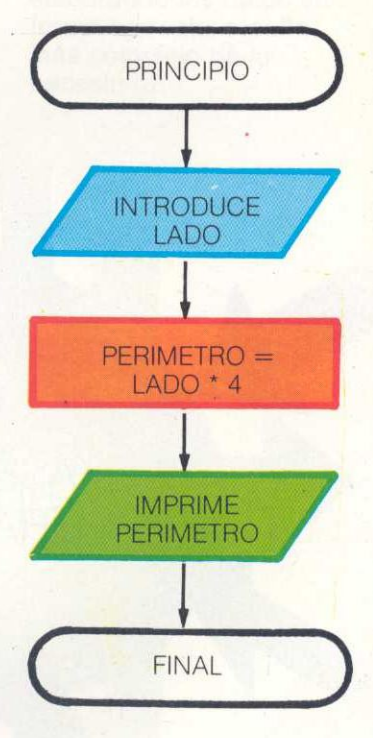
DECISION

ENTRADA
SALIDA

lenguaje máquina o Assembler) el trabajo a desarrollar es notable.
Por ejemplo, es especialmente difícil encontrar un procedimiento general y definido allí donde no lo haya. Con este objeto ha sido introducido el llamado diagrama de flujo.
Este, como se puede intuir por el término, nos permite ver a través de un esquema como se desarrolla el algoritmo, es decir, cuál es su flujo.
Cada una de estas figuras, unidas entre ellas por una línea con flecha que indica la dirección del flujo, contiene una instrucción del algoritmo.
Concluyendo: un algoritmo puede representarse mediante un diagrama de bloques.
Hay otros símbolos además de los cuatro mostrados, pero estos son por ahora más que suficientes.
Sin embargo, cuando se habla de solucionar un problema por este método, oiréis decir que se está programando ese problema y no haciendo un algoritmo.
¿Por qué? La diferencia es sutil pero importante.
Hemos dicho que un algoritmo sirve para resolver un problema, que nos da los medios para dar una respuesta.
¿Pero nos basta para hacer resolver un problema a una máquina? La respuesta es no. El algoritmo es básico porque indica a la máquina cómo resolver el problema, pero nosotros tenemos que «organizarlo» para que dé la respuesta correcta.

Escribir el programa

Supongamos que en un programa hemos incluido el procedimiento para transformar millas en kilómetros, pero que sin embargo nos interesa que un cálculo sea expresado en millas: ¿qué haremos?: sencillamente, excluir el algoritmo de conversión cuando no nos convenga que opere. Pero, ¿esto quién lo ordena?: el programa.
Asi pues, el programa es una serie ordenada de algoritmos organizada de forma que defina su propia secuencia (el orden de ejecución).
Ahora vamos a examinar los diversos pasos de la realización de un programa.
Hagamos un simple programa para calcular el perimetro de un cuadrado cualquiera.
Al principio el programa deberá pedirte medida del lado, y finalmente te mostrará el valor del perímetro.
Básicamente puede decirse que te pide una información de entrada (introduces el lado) y te proporciona un resultado (perímetro).
En general, dado que un programa es una sucesión de algoritmos, este tendrá que tener al menos un dato de entrada y otro de salida.
Normalmente este proceso, definido como de entrada/salida (input/output), es la primera fase de la solución de un problema a tavés de un ordenador (es decir,



hay que definir mediante qué variables se puede resolver el problema). La fase siguiente es el "cómo" usar esas variables, es decir hay que escribir el algoritmo.
La tercera y última fase es encontrar una forma conveniente para la salida de los datos (output). Si resuelves estos puntos podrás programar cualquier problema.
Es aconsejable (aunque no necesario) analizar el problema en el orden expuesto, de forma que todos los pasos a realizar estén ligados entre ellos, facilitando así la puesta a punto del programa.
Por lo tanto, en el programita de antes, debes establecer:
1) ¿Cuántas y cuáles variables necesitamos? Tienes que procurar que el programa pida los datos necesarios (en este caso uno solo: el lado).
2) ¿Qué procedimiento usar? En este caso un procedimiento que te permita calcular el perimetro (lado x 4).
3) ¿Cómo presentar el resultado? Imprimiéndolo en pantalla.
Como puedes ver, el problema no es aprender sobre qué instrucciones se basa el funcionamiento de un ordenador, sino que se trata de reducir tu objetivo de tal forma que lo hagas "digerible" para el ordenador.
Por lo tanto, programar, no significa solamente el perfecto conocimiento de lo que puede ofrecer la máquina, sino que también implica una profunda comprensión del problema a afrontar.



Así pues, no sólo es necesario un conocimiento "informático", estrictamente ligado a los ordenadores, sino también un conocimiento de los métodos que es útil usar para reducir el problema a términos aceptables para el ordenador.
Para confeccionar buenos programas es necesaria por lo tanto una forma de proceder analitica, capaz de resolver un problema en forma lógica.
Esta es la habilidad indispensable para un buen programador.
Conviene advertir que el diagrama de flujo como método para resolver un problema, ha sido suplantado por la programación estructurada.
Cuanto más complicado sea el problema, más se complica el diagrama de flujo que representa su algoritmo.
Esto comporta obviamente varios inconvenientes, pero lleva sobre todo a tener un programa poco legible o difícilmente modificable.
De cualquier manera el diagrama de flujo sigue siendo un método óptimo para ilustrar lo que el programa es capaz de hacer, aunque por si solo sea insuficiente para la realización de un programa.
Recapitulando:

- Usa el diagrama de flujo para ilustrar un programa, y no para escribirlo;
- Piensa mediante modelos y crea un procedimiento general lógico válido para resolver el problema;
- Utiliza los diagramas de bloques para corregir programas demasiado tortuosos: por el dibujo podrás ver facilmente si el programa, aunque elaborando los datos en forma correcta, resulta más complejo de lo necesario.



Tu primer programa

Hace poco que estudias BASIC y programación conmigo, pero ya tienes todos los elementos para escribir un programa sencillo, pero efectivo.
Te guiaré paso a paso aunque pronto te darás cuenta de que esto ya no es tan necesario.
Escribamos juntos un programa que calcule el área de un triángulo cualquiera.
Como ya se ha dicho más veces, tu Spectrum unicamente puede elaborar datos que tu le proporcionas.



Fase I
Por lo tanto, la primera fase es la de localizar los datos necesarios para el cálculo del área de un triángulo, e introducirlos.
Como bien sabes, se necesita conocer la base (B) y la altura (H); para tu ordenador estas serán variables numéricas reales.



Fase II
Después de que el Spectrum haya recibido estas informaciones, debes instruirle sobre qué operaciones realizar. Hemos llegado a la segunda fase.
Nuestro programa se propone como objetivo el cálculo del área de un triángulo (S) que se puede obtener empleando la siguiente fórmula:

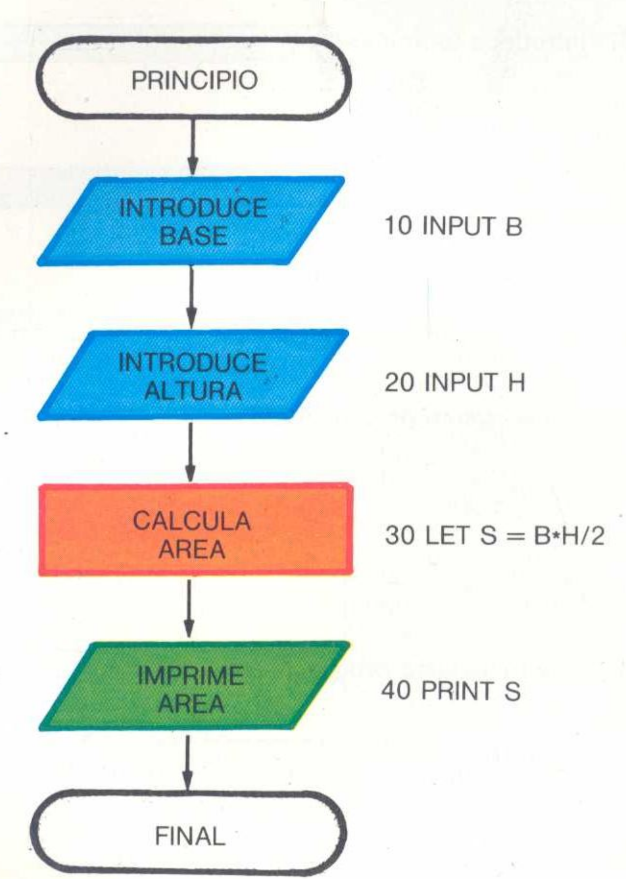
$$S = \frac{\text{Base x Altura}}{2} = \frac{\text{B x H}}{2}$$

Debes, por tanto, proporcionar esta fórmula a tu ordenador para que pueda proceder a la elaboración de los datos.



Fase III

Tu Spectrum posee ahora todos los elementos para proceder al cáculo del área de un triángulo. Pero el resultado se lo queda para él, si no le dices cómo mostrátelo. disponible.
Hemos llegado a la última fase, la de SALIDA, en la que el ordenador debe devolverte el resultado de las elaboraciones, es decir, el objetivo del programa.
Traduzcamos ahora todo lo dicho en palabras a un diagrama de bloques junto a las instucciones BASIC que conoces.



No ha sido muy difícil ¿verdad?

EJERCICIOS

Anota en el espacio en blanco el resultado que preveas para cada ejercicio propuesto, y verifica después la solución con tu Spectrum. Si has cometido tan sólo un error repasa la lección.

PRINT "Introduce un número";: INPUT N

INPUT "Introduce un número"; N

```
10 PRINT "TU MICRO"
20 PRINT "PERSONAL SOFTWARE"
30 PRINT "VIDEOJUEGOS"
```

Para el programa precedente LIST 20

Sigue para el mismo programa LIST

Todavia para el mismo programa RUN 20

Y ahora RUN

SEIKOSHA SP-800

El fruto de la Investigación

La nueva impresora de **SEIKOSHA SP-800**, con un ordenador personal puede escribir ***96 combinaciones de letra diferentes***, desde 96 caracteres por segundo a 20 con muy alta calidad de letra, además es gráfica en alta densidad.

Su precio es de 69.900 Pts con ***introductor automático hoja a hoja***.

Con un pequeño ordenador personal, un procesador de textos puede costar alrededor de cien mil pesetas.

Infórmese y comprenderá por qué **las máquinas de escribir** tienen demasiados años.

Nuestra calidad es "SEIKO";
nuestros precios, únicos
Si desea más información,
consulte con nuestro distribuidor
más cercano, llame o escriba a:



DIRECCION COMERCIAL:
Av. Blasco Ibañez, 114-116
46022 VALENCIA
Tel.(96) 372 88 89
Télex 62220
DIRECCION COMERCIAL EN CATALUÑA:
C/Muntaner, 60-2-4Pta
08011 BARCELONA
Tel.(93) 323 32 19

ESTOS SON NUESTROS MODELOS:

MODELO		VELOCIDAD	COLUMNAS	TIPOS DE LETRA	P.V.P.R * INTERFACE PARALELO
GP-50	LA PEQUEÑA	40 cps	46	2	25.900
GP-500	LA ECONOMICA	50 "	80	2	47.900
GP-550	LA STANDARD	86 "	80-136	18	59.900
SP-800	LA PERFECCION	96 "	80-137	20	69.900
GP-700	LA DE COLOR	50 "	80-106	3	84.900
BP-5200	LA DE OFICINA	200 "	136-272	18	199.900
BP-5420	LA MAS RAPIDA	420 "	136-272	18	299.900

* Los precios indicados son los recomendados para conexión tipo paralelo Centronics, para otro tipo de conexión, sufren un ligero incremento.

Este pie de página ha sido realizado integramente con la nueva impresora:

SEIKOSHA SP-800